

# Frequently Asked Questions About Sampling on Linux\*

## Using the VTune™ Performance Analyzer

Rev 0.23

---

### Topics

1. [Overview of Remote Sampling](#)
2. [Sampling Over a Single Linux\\* Process](#)
3. [Requirements on Linux Binary for Hotspot Drill Down](#)
4. [Requirements on Linux Binary for Source Drill Down](#)
5. [Specifying Environment Variables](#)
6. [Requirements for Launching an Application](#)
7. [Using Special Characters >, <, |, &, etc.](#)
8. [Running vtserver as Root](#)
9. [Maximum Time Possible for Sampling](#)
10. [Factors that Determine Amount of Sampling Data Generated](#)
11. [No Sampling Data Generated](#)
12. [No Sampling Data Collected for Process or Module](#)
13. [Meaning of "OtherNN" in Module View](#)

- 14. [Specifying Binary Files for Hotspot Drill Down](#)
- 15. [Specifying Source Files for Source Drill Down](#)
- 16. [Source Hotspots in Wrong Location](#)
- 17. [Sampling Driver Errors](#)
- 18. [Multiple Users Sampling Simultaneously](#)
- 19. [Restarting vtserver Fails](#)
- 20. [Automating and/or Batch Processing Runs](#)

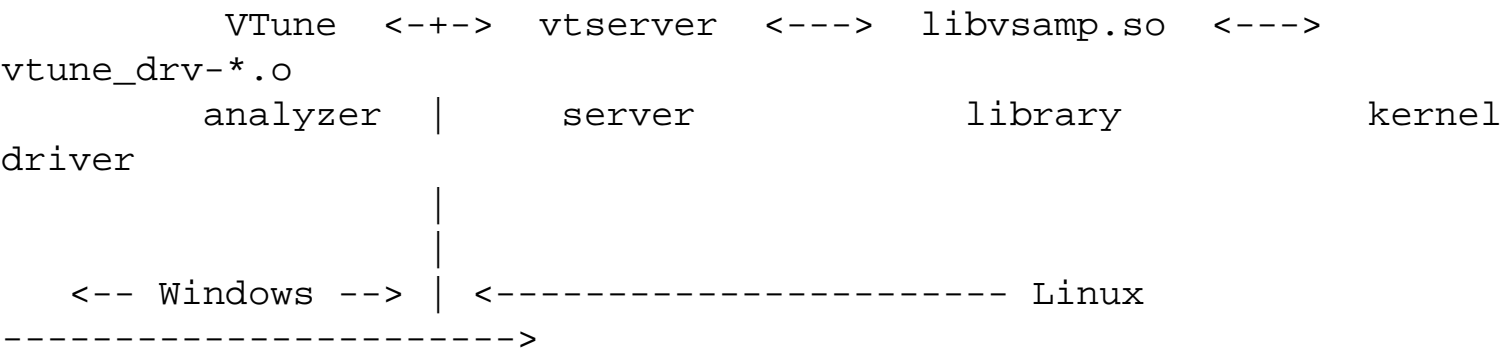
# Legal

## [Disclaimer and Legal Information](#)

## 1. Overview of Remote Sampling

*How does remote sampling work?*

For remote sampling, two systems are required: one running the VTune™ Performance Analyzer on a Microsoft\* Windows\* operating system, and another running remote data collectors on a Linux\* operating system. The diagram below shows the relationship between components on these systems.



The kernel driver, vtune\_drv-\*.o, is first loaded into the kernel by the root user. This must be done prior to starting the server, vtserver. The kernel driver is responsible for configuring and reading the

hardware performance counters.

When the server first starts, it initializes all of its collectors. Initializing the sampling collector involves loading the sampling library, `libvsamp.so`. The sampling library is a user-level interface to the kernel driver, and supports several functions including configuring, starting, and stopping the kernel driver.

The VTune analyzer sends a message to the server on a specified port and waits for a reply. The server dispatches that message to the appropriate collector, and then continues listening for more requests from the VTune analyzer. When the collector finishes, it notifies the server, which in turn notifies the VTune analyzer. If requested, the server also sends data resulting from the task back to the VTune analyzer. The analysis and presentation of this data is then done in the VTune analyzer.

## 2. Sampling Over a Single Linux Process

*Is there a way to perform sampling on just one Linux process?*

No. Since sampling is system-wide, any kernel or user-level process that runs during the collection may have sampling data associated with it. However, after sampling completes, you can use the VTune analyzer to drill down on the particular Linux process (or modules within that process) you are interested in.

## 3. Requirements on Linux Binary for Hotspot Drill Down

*What are the requirements on a Linux binary in order to drill down to hotspots and view function names?*

In order to drill down to hotspot and view by function, the Linux binary must have symbols. For example, if the binary is an executable, then it should not be stripped (this can be checked by using the `nm` command on the executable). If the binary is a shared library, then it must have at least dynamic symbols (this can be checked by using the `objdump` command on the library). If the binary is a Linux kernel, then the kernel file must have symbols and be in non-compressed, non-bzImage format.

## 4. Requirements on Linux Binary for Source Drill Down

*What are the requirements on a Linux binary in order to drill down to its source?*

In order to drill down to source, the Linux binary must be compiled with debug information. For example, if the GCC or the Intel(R) compilers are used, then the sources should be compiled with the "`-g`" option. Please note that under certain circumstances, compiling with detailed debug information and high optimization (e.g., "`-g3 -O3`") may cause problems with source drill down.

In order to drill down to Linux kernel source, you must have a kernel file built with debug information. To build such a kernel file, go to the directory containing the kernel sources and edit the kernel source's top-level `Makefile`. Add the "-g" option to the following variables:

```
...
# Include the make variables (CC, etc...)
...
CFLAGS_KERNEL = -g
...
# standard CFLAGS
...
CFLAGS := ... -g ...
...
```

Then run "make clean all" to create the `vmlinux` kernel file with debug information. After performing a sampling session, specify that file as the one to drill down on in the VTune analyzer Hotspot View.

## 5. Specifying Environment Variables

*How do I specify environment variables to pass to an application when it gets launched?*

There are several ways to pass environment variables to your application. Select whichever approach best meets your needs.

- Set the environment variable in the shell where `vtserver` starts from, and then start `vtserver`. All environment variables known to `vtserver` will be passed to the launched application.
- When configuring the application to be launched in the VTune analyzer, go to the Application/Module Profile dialog box, select the "Advanced .." button. Uncheck the "Use default environment" option and then enter in lines of the form `VARIABLE=VALUE` . Note: variables specified here will override matching ones inherited by the application from `vtserver`'s environment.
- Create a shell script which sets the variables and then runs the application. Then specify that shell script as the application to launch in the VTune analyzer.

## 6. Requirements for Launching an Application

*In the VTune analyzer, I specify a Linux application to launch, but the application does not get executed. Why is that?*

The following conditions must hold in order to launch a Linux application:

- the application must reside in a location that is accessible by the `vtserver` process
- the application must be executable by the `vtserver` process
- the application cannot be a command-line application that expects interactive user input
- if the application is a script, then the first line of the script must be of the form `"#!/path/to/interpreter ..."`

## 7. Using Special Characters `>`, `<`, `|`, `&`, etc.

*When I use special characters such as `>`, `<`, `|`, or `&`, as part of the command-line arguments of the application to launch, the VTune analyzer fails to launch the application. Why is that?*

Special characters, such as `>`, `<`, `|`, and `&` are not arguments to the application itself, but rather are used in the context of a shell interpreter (e.g., `bash`, `csh`, `tcsh`, etc.). If such characters are required, then create a shell script which invokes the actual program and uses those characters. Then specify that script as the application to launch.

## 8. Running `vtserver` as Root

*I'm running `vtserver` as root, but `vtserver` has problems accessing files. Why is that?*

If the files are located on a networked filesystem (e.g., through NFS\*, AFS\*, or Samba\*), then there may be restrictions on root access to those files (or directories containing those files) on that filesystem. Check with your system administrator for details.

Note: it is highly recommended that `vtserver` be run as a *non-root* user. If multiple users need to use `vtserver`, then add those users to a special group and make sure the relevant files and directories can be accessed by members of that group. Then run `vtserver` as one of the non-root members of that group.

## 9. Maximum Time Possible for Sampling

*Is there a limit to how long I can sample?*

The maximum time a sampling collection can run is limited only by the amount of disk space available on the filesystem where sampling data (`*.tb5`) is being stored. In general, sampling should be stopped when the size of the sampling data reaches half the free space left on the filesystem. For example, on the Linux side, if the Linux filesystem system has 100 MB free, then sampling should be stopped when the size of the sampling data reaches about 50 MB. Similarly, on the VTune analyzer side there must be enough space in the `Projects\directory` to hold the transferred sampling data plus a generated database of approximately the same size as the sampling data.

## 10. Factors that Determine Amount of Sampling Data Generated

*What factors determine how much sampling data gets generated?*

The amount of sampling data collected depends on several factors:

1. the **duration** of the sampling session
2. the **number of events** to collect samples over
3. the **frequency of an event** being sampled; for example, clocktick events occur every cycle, whereas L2 cache miss events occur less frequently (or not at all), depending on the processes running during the collection
4. an event's **sample after value (SAV)**; the SAV determines the rate at which samples are collected, e.g., after SAV number of events occur, an interrupt occurs and one sample for that event is taken; keep in mind that,
  - specifying a low SAV will yield potentially large number of samples
  - specifying a high SAV will yield potentially low (or zero) number of samples
  - the VTune analyzer initially assigns a  $SAV = CPU\text{speed}/1000$  for the default clockticks and instructions retired events; the default SAV for other events are fixed in the processor XML files used by the VTune analyzer
  - if calibration is turned on, then the VTune analyzer will adjust the SAVs based on the **sampling interval (SI)**; for example, if SI is 1 millisecond, then an event's SAV will be adjusted so that approximately 1000 samples can be obtained per processor per second during the collection; in general, an event's SAV is calculated so that  $(\#samples\ per\ processor/second) * SI = 1$
  - there is a performance tradeoff: the higher the sampling resolution (i.e., lower the SAV) the more overhead will be incurred by the sampling process, and the less work your application (or any application) will do during the collection
5. the **number of CPUs** detected by the OS and used during the collection

## 11. No Sampling Data Generated

*I ran a sampling session, but the VTune analyzer reports no data was generated. Why is that?*

It is possible that for the selected events, either those events did not occur, or the SAVs for those events were set too high.

## 12. No Sampling Data Collected for Process or Module

*I ran a sampling session, but my applicatoin does not appear in the VTune analyzer Process View or Module View. Why is that?*

It is possible that either the SAVs for the selected events were set too high, or your application did not exercise code that would trigger the selected events to occur.

## 13. Meaning of "OtherNN" in Module View

*Where did the "Other16", "Other32", or "Other64" modules shown in Module View come from? And why are there samples associated with them?*

The OtherXX does not represent an actual module, but rather a category for placing samples that the VTune analyzer was unable to associate modules with. The XX in OtherXX means that when the sample was collected, the associated processor was operating in XX addressing mode (e.g. Other32 indicates the processor was operating in 32-bit addressing mode).

## 14. Specifying Binary Files for Hotspot Drill Down

*When drilling down to hotspot on a Linux binary, the VTune analyzer prompts for the binary file's location. Why is that?*

Currently, symbol processing and analysis of Linux binaries is done in the VTune analyzer on the Windows side. When the VTune analyzer encounters a reference to a Linux binary for the first time, it will prompt for the location of where to retrieve the file. There are several ways the binary file can be retrieved:

- through vtserver: the file location must be of the form `/path/to/the/binary`, and must be accessible by the vtserver process
- manually copied from Linux to Windows: the file location should follow the Windows filename syntax, e.g., `DRIVE:\path\to\binary`
- remote Linux file system exported through NFS, AFS, Samba, etc.: the file location should follow the appropriate Windows filename syntax, e.g., `\\linuxMachineName\path\to\binary` (if not mapped to a `DRIVE:` letter), or `DRIVE:\path\to\binary`; contact your system administrator for details

Once the VTune analyzer processes the binary, the binary's location will be cached. Subsequent drill downs on the same binary will re-use the cached location. If the VTune analyzer detects that the drilled down binary has changed, or if the cached location for that binary has been cleared, then the VTune analyzer will prompt again for the location of where to retrieve the binary file.

Note: the cached location can be cleared by going to the `Configure->Options->Directories->Symbol Repository->View Symbol Repository Files` dialog box in the VTune analyzer.

## 15. Specifying Source Files for Source Drill Down

*When drilling down to source on a Linux binary, the VTune analyzer prompts for the location of the source file. Why is that?*

Currently, the matching of sampling data to source is done in the VTune analyzer on the Windows side. When the VTune analyzer encounters a reference to a Linux source file for the first time, it will prompt for the location of where to retrieve the file. There are several ways the source file can be retrieved:

- manually copied from Linux to Windows: the file location should follow the Windows filename syntax, e.g., `DRIVE:\path\to\sourceFile`
- remote Linux file system exported through NFS\*, AFS\*, Samba\*, etc.: the file location should follow the appropriate Windows filename syntax, e.g., `\\linuxMachineName\path\to\sourceFile` (if not mapped to a `DRIVE:` letter), or `DRIVE:\path\to\sourceFile`; contact your system administrator for details

Once the VTune analyzer matches the source against the sampling data, the source location will be cached. Subsequent drill downs on the same source will re-use the cached location. If the VTune analyzer detects that the drilled down binary has changed, or if the cached source location for that binary has been cleared, then the VTune analyzer will prompt again for the location of where to retrieve the source file.

Note: the source location cache can be cleared by going to the `Configure->Options->Directories->File Association` dialog box in the VTune analyzer.

## 16. Source Hotspots in Wrong Location

*When I drill down to Source View, the source lines where the samples occur seem off from what I expect. Why is that?*

Some Intel processors provide special event registers that hold the exact instruction being executed when an event occurs.

On Pentium® 4 processor-based systems, events which are "precise" can be associated with the exact instruction being executed when the event occurs.

Similarly on Itanium®-based systems, "EAR" events can be associated with the exact instruction being executed when the event occurs.

For all other events, the next instruction executed after the counter overflow interrupt occurs is captured and associated with that event. Since the VTune analyzer cannot adjust for these kinds of events, samples may appear on source lines later than expected.



Search under "event skid" in the VTune analyzer online help for further information.

## 17. Sampling Driver Errors

*The server reports a "LIBVSAMP Error: . . ." occurred during a sampling collection. What should I do?*

When a sampling driver error is reported, it is recommended that the following steps be taken:

1. log onto your Linux system as root
2. unload the sampling driver using the `rmmod-vtune` utility
3. re-load the sampling driver using the `insmod-vtune` utility
4. re-start the server, `vtserver`, to begin a new sampling run

## 18. Multiple Users Sampling Simultaneously

*Can several users perform sampling at the same time on a given Linux system running vtserver?*

No. Although multiple users can connect to `vtserver`, only one sampling session may be active at a time. Users who wish to run individual sampling sessions on the same Linux system should coordinate amongst themselves to ensure their sampling sessions do not overlap.

## 19. Restarting vtserver Fails

*When I shutdown vtserver using Ctrl-C, and then try to immediately restart it, vtserver reports that it's unable to bind to port 50000. Why is that?*

The server, `vtserver`, communicates with the VTune analyzer through a socket connection on port 50000. When `vtserver` and all applications launched by `vtserver` are shut down, the socket connection on the Linux side may take some time to completely close. You can check the status of the connection via the command,

```
netstat -a | grep tcp | grep :50000
```

When the connection closes (i.e., is no longer listed by the above command), then you may restart `vtserver`.

## 20. Automating and/or Batch Processing Runs

*Is there a way to automate or batch process the collection and analysis of sampling data?*

The VTune analyzer provides a "batch mode" API that allows programs or scripts running on Windows operating systems to automatically control the VTune analyzer. Search under "batch mode" in the VTune analyzer online help for further information.

There are also 3rd-party automation and scripting Windows tools, which could be used to simulate user inputs to the VTune analyzer and perform scripting of these runs. Examples of such tools include [AutoIt\\*](#), [AutoMate\\*](#), and [QuickMacros\\*](#).

Automation on the Linux side could be done remotely using the aforementioned tools in conjunction with Windows-based SSH clients such as [PuTTY\\*](#).

---

## Disclaimer and Legal Information

Information in this document is provided in connection with Intel products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, or life sustaining applications.

This *FAQ document for Sampling* as well as the software described in it is furnished under license and may only be used or copied in accordance with the terms of the license. The information in this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Intel Corporation. Intel Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in this document or any software that may be provided in association with this document.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

Intel SpeedStep, Intel Thread Checker, Celeron, Dialogic, i386, i486, iCOMP, Intel, Intel logo, Intel386, Intel486, Intel740, IntelDX2, IntelDX4, IntelSX2, Intel Inside, Intel Inside logo, Intel NetBurst, Intel NetStructure, Intel Xeon, Intel XScale, Itanium, MMX, MMX logo, Pentium, Pentium II Xeon, Pentium III Xeon, and VTune are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

\*Other names and brands may be claimed as the property of others.

Copyright© Intel Corporation 2000 - 2003.